REMARKS

Claims 9-11 have been cancelled. Claims 12 – 19 , 28, and 30 – 34 remain pending and have been amended as noted below. New claims 35-40 have been added.

I. 35 U.S.C. §102 Rejections

In the Office Action mailed March 22, 2010, all currently-pending claims were rejected as anticipated by U.S. Patent Application Publication No. 2006/0200535 (Moser). As noted previously, anticipation under 35 U.S.C. § 102 requires that the applied reference teach the identical invention in which all of the claimed limitations are arranged or combined in the exact same way as recited in the claim.

1. Moser does not teach use of the same computing device to generate/create the descriptor tree and to instantiate and to render application elements based on the descriptor tree

Each of independent claims 12, 28, and 32 as amended, along with new claim 38 clearly indicates that the same computing device/processor creates the descriptor tree and renders interface elements of the application. Additionally, the claims indicate that a graphical representation is rendered using the instantiated objects. See, e.g., paragraph [0023] of the specification ("...the actual object, which the graphical player *uses to render the graphical representation* on its display...").

Moser does not teach use of the same computing device for these purposes. Instead, the Office Action relies on portions of Moser describing server-side and client-side activity. For instance, server-side activity is alleged to teach generating the descriptor tree (see, e.g., the Office Action's citation to paragraph [0083], which describes the *server*-renderer 101-2

generating a component descriptor from model 200-T1). The Office Action relies on paragraph [0135] as teaching the previously-claimed rendering. Without agreeing as to whether that is proper, Assignee notes that paragraphs [0134]-[0135] describe use of the DOM 100-9 at the *client* to provide the visualization of the page ("DOM 100-9 is used to present the graphical user interface of an application to the user"). DOM 100-9 is different from the server-side model 101-2 alleged to teach other claim elements. At least because the portions of <u>Moser</u> alleged to teach the claimed elements are not arranged in the same manner as in the claims, <u>Moser</u> does not anticipate.

Assignee notes that <u>Moser</u> also teaches some client-only activity, but this appears to be based on caching browser-increments (see paragraph [0131]) and not by generating or creating a descriptor tree at the client based on code of the application.

2. *Moser does not teach use of the descriptor tree to determine which interface elements to instantiate*

Claim 12 indicates that <u>the graphical representation [is] rendered by using the descriptor tree to identify</u> corresponding ones of said plurality of descriptor nodes and said one or more stacked descriptor nodes. Claim 28 already indicates that identifying comprises <u>using</u> the descriptor tree to identify a node whose object is for rendering an interface element that is not visible in the initial view and (ii) excluding the object from the subset of objects. However claim 28 further indicates that instantiating is <u>in response to identifying the subset of objects</u>. Claim 32 indicates that instantiating is based on <u>identifying</u> corresponding non-hidden nodes <u>in the descriptor tree</u>. New claim 38 indicates that the processor is configured to *use the descriptive hierarchical structure to identify* the at least one first descriptor node and instantiate an object corresponding to the at least one first descriptor node

The Examiner relies on portions of Moser that relate to changing the visualization of the tree 955. Although Moser refers to a description of a tree and to "descriptors," Assignee does not believe that Moser teaches a descriptor tree. However, the claims have been clarified to explicitly note that the descriptor tree is used in order to determine which elements/objects to instantiate or to render. In contrast, the Moser browser fragments (alleged to teach the descriptor tree) are not analyzed to determine whether or not to instantiate an object. Instead, Moser uses the browser fragments to modify the DOM (which is then used to render the interface).

3. *Moser does not teach use of descriptor nodes that describe an interface element not shown on the currently visible pane*

Claims 12, 28, and 32 refer to inclusion of nodes in the descriptor tree describing unseen elements/objects not instantiated at the start of the application. New claim 38 refers to at least one "first" descriptor node that corresponds to an object used in rendering in an initial view of the application and at least one "second" descriptor node that corresponds an object used in rendering a second view of the application other than the initial view of the application

The Examiner relies upon generation of HTML or other browser-compatible code in Moser as teaching the "descriptor tree." The Examiner alleges that the component descriptors of the tree meet the claimed "plurality of descriptor nodes" that describe an interface element currently instantiated and visible.

Use of the iFrame element (referenced in the rejection of claim 9) relates to retrieving the browser increments, and not to providing browser increments related to uninstantiated elements, so Assignee does not believe that portion of Moser is on point. However, the rejections also refer to different states of tree 955, including some in which the tree is collapsed. Assignee respectfully asserts that this does not teach description of an interface element or object that is

not instantiated or used in an initial view. Even if a an element of the <u>Moser</u> tree is collapsed, the corresponding interface element (e.g., the element used to render the (+) symbol) is instantiated or used. In contrast, the claims refer to a descriptor node for elements/objects that are not instantiated/used at all.

Additionally, the descriptor nodes are included in the same descriptor tree; in <u>Moser</u>, the browser increments are not part of the same structure. New claim 40 has also been added to explicitly note that the descriptive hierarchical structure includes objects of a plurality of different views.

4. *Moser does not teach use of nodes containing software methods*

Claim 14 was rejected because <u>Moser</u> allegedly teaches use of a script tag at paragraph [0110]. Assignee respectfully notes that paragraph [0110] refers to *using* the script tag to retrieve the browser increment 300-I, and paragraphs [0111]-[0017] states that the client-side portion of the framework uses the script tag in the DOM so that the client sends a server request to the server. In contrast, claim 14 refers to content of the *descriptor tree* as including a constructor method.

New claims 35-37 indicate that (1) the descriptor tree comprises a hierarchical data structure and (2) that the tree comprises an application descriptor node including a child constructor method for creating a child node of the application descriptor node, the child constructor method comprising code logic containing a pointer to code of the rich internet application. Support can be found in the specification, including at page 7, paragraph [0024]. Assignee respectfully asserts that these features are not found in <u>Moser</u> as arranged in the claim.

II. Conclusion

Assignee respectfully asserts that the claims are more than a mere variation on the <u>Moser</u> system and that the subject matter would not have been obvious under 35 U.S.C. §103. In particular, <u>Moser</u> is directed to a client-server architecture. The browser increments are generated based on modifications to the model (e.g., model 200-T1 discussed in paragraphs [0078]-[0083]), which is maintained at the server, in order to reduce bandwidth consumption and CPU time that would be associated with generating and sending a representation of an entire page to the browser (see paragraph [0125]). Changing <u>Moser</u> to operate on a single computing system would go against the principles of operation of render <u>Moser</u> unsuitable for its intended purpose (client-server communications). Removing the <u>Moser</u> components for use in a standalone application would appear to be based purely on hindsight, and thus would make for an improper basis for rejecting the claims under §103.

In view of the foregoing, Assignee respectfully asserts that the subject matter of the independent claims should be allowable. The dependent claims should be allowable for at least the same reasons. No fee is believed due with this response. However, if a fee is due, please charge our Deposit Account No. 11-0855.

Respectfully submitted,

Date: June 22, 2010

/Eric G Zaiser/

Eric G. Zaiser
Reg. No. 58,352

KILPATRICK STOCKTON LLP
1100 Peachtree St. NE
Atlanta, GA 30309
404.745.2447 (voice)
404.541.3290 (fax)

US2008 693072.1